Videojuegos

Curso de Diseño y Programación

Nº 6 5,99 euros

Diseñar los **menús** y los **tutoriales** del juego

Manejar texto e imágenes en **Blitz 3D**

Cómo cuantizar con **Cubase VST**









Curso práctico de Videojuegos

AUTOR DE LA OBRA
Marcos Medina
DIRECCIÓN EDITORIAL
Eduardo Toribio
etoribio@iberprensa.com
COORDINACIÓN EDITORIAL
Eva-Margarita García
eva@iberprensa.com
DISEÑO Y MAQUETACIÓN
Antonio Gª Tomé
PRODUCCIÓN
Marisa Cogorro
SUSCRIPCIONES
Tel: 91 628 02 03
Fax: 91 628 09 35
suscripciones@iberprensa.com
FILMACIÓN: Fotpreim Duvial
IMPRESIÓN: Gráficas Don Bosco
DUPLICACIÓN CD.ROM: M.P.O.
DISTRIBUCIÓN
S.G.E.L.
Avda. Valdelaparra 29 (Pol. Ind.)
28108 Alcobendas (Madrid)
Tel.: 91 657 69 00

EDITA: Iberprensa
www.iberprensa.com



CONSEJERO
Carlos Peropadre

REDACCIÓN, PUBLICIDAD Y ADMINISTRACIÓN
C/ del Río Ter, 7 (Pol. Ind. "El Nogal")
28110 Algete (Madrid)
Tel.: 91 628 02 03
Fax: 91 628 09 35
(Añada 34 si llama desde fuera de España.)

DEPÓSITO LEGAL: M-35934-2002
ISBN: **Coleccionable:** 84 932417 2 5
Tomo 1: 84 932417 3 3
Obra Completa: 84 932417 5 X

NOTA IMPORTANTE:
Algunos programas incluidos en los CD de "Programación y Diseño de Videojuegos" son versiones completas, pero en otros casos se trata de versiones demo o trial, versiones de evaluación que Iberprensa quiere ofrecer a nuestros lectores. No se trata en ningún caso de las versiones comerciales de los programas, y las hemos incluido para dar al lector la oportunidad de conocer y probar esos programas y que así pueda decidir posteriormente si desea o no adquirir las versiones comerciales de cada uno.

Aprende divirtiéndote

Bienvenidos a **Programación y Diseño de Videojuegos**, la primera obra coleccionable cuyo objetivo es formar al alumno en las principales técnicas relacionadas en el desarrollo completo de un videojuego.

A lo largo de la obra el lector aprenderá programación a nivel general y a nivel específico con ciertas herramientas y lenguajes, aprenderá a trabajar con aplicaciones de retoque de imagen y también de diseño 3D y animación. Descubrirá las aplicaciones profesionales más importantes de audio y conocerá la historia de lo que se denomina "la industria del videojuego", los últimos 20 años, los juegos que marcaron un avance, sus creadores y en general la evolución del videojuego.

Pero además, esta obra tiene un segundo objetivo, desarrollar y potenciar la creatividad del lector, nosotros a lo largo de las diferentes entregas pondremos las bases y tú pondrás tu ingenio, tu creatividad y tu capacidad de mejorar.

Comienza aquí un viaje de 20 semanas articulado en 400 páginas y 20 CD-ROMs cuya finalidad es proporcionar las bases mínimas para después cada uno continuar su camino.

Recuerda que para alcanzar el éxito necesistas cumplir tres condiciones: que te gusten los juegos, poseer cierta dosis de creatividad y finalmente capacidad de estudio.

Una la cumples seguro.

sumario



PARA ENCUADERNAR LA OBRA:

- *Para encuadernar los dos volúmenes que componen la obra "Programación y Diseño de Videojuegos" se pondrán a la venta las tapas 1 y 2.*
- ***Tapas del volumen 1 ya a la venta.***
- ***Los suscriptores recibirán las tapas en su domicilio sin cargo alguno como obsequio de Iberprensa.***

SERVICIO TÉCNICO:
Para consultas, dudas técnicas y reclamaciones Iberprensa ofrece la siguiente dirección de correo electrónico: games@iberprensa.com

PETICIÓN DE NÚMEROS ATRASADOS:
El envío de números sueltos o atrasados se realizará contra reembolso del precio de venta al público más el coste de los gastos de envío. Pueden ser solicitados en el teléfono de atención al cliente 91 628 02 03

Diseño de los elementos de un juego III: presentación, menús y tutoriales

En la anterior entrega diseñamos el entorno del juego con todos sus elementos. Nos queda plantear los menús, el aspecto que tendrá la pantalla de juego y algunos de los elementos necesarios para la correcta terminación de nuestro trabajo, como la presentación y los tutoriales de ayuda.

(■) PRESENTACIÓN

Todo juego tiene que tener una presentación por pequeña y simple que sea. Generalmente, se compone de varias partes que van desde mostrar los logotipos del distribuidor y desarrollador hasta el argumento del juego. Se pueden realizar presentaciones que sean verdaderas películas de animación o simplemente mostrar una pantalla fija con los logos y un texto argumentativo. La cuestión es que el futuro jugador sepa a qué clase de aventura se enfrenta. Generalmente, el argumento del juego se muestra en esta parte y aunque su finalidad sólo sea ésa, cumple un papel muy importante en la captación del interés del usuario, que tiene que perder algún tiempo antes de poder jugar. Debido a esto, es conveniente que la presentación pueda ser cortada a voluntad haciendo clic con el ratón o pulsando una tecla.



Un ejemplo del flujo en una presentación aplicada a "Zone of Fighters".

(■) LOGOTIPOS

Pero centrémonos en cada una de las partes que forman la presentación, empezando por los logotipos. Estos elementos sirven para colocarle al juego la "etiqueta de origen". Se muestran los logotipos de quien lo distribuye o de quien lo ha realizado y por último el propio logo o título del juego. Hay que tener en cuenta que el orden de aparición es importante. En primer lugar debe aparecer el distribuidor, seguidamente el desarrollador o entidades colaboradoras y por último la presentación del juego y su argumento.

(■) MOSTRAR EL ARGUMENTO

Por lo general, existe una clara división entre presentar a los artífices del juego y la descripción de éste. La mayoría de las distribuidoras y desarrolladoras tienen su forma de mostrarse al público, bien con una animación de su logo o con una pantalla fija y en muchas ocasiones utilizan diferentes métodos gráficos. Lo que sí tenemos que darle importancia es a la presentación del juego en sí, es decir, de su argumento. Desde un cortometraje de animación hasta un texto, todo vale, pero la calidad de esta parte del proceso dará una impresión mejor del juego. Lo primordial es contar de qué va. Dónde se desarrolla la acción, quiénes son los malos y los buenos y qué misión debe cumplir el jugador. Actualmente se realizan verdaderas joyas de la animación para presentar un juego y que se utilizan también como reclamo publicitario en su comercialización, al igual que se hace con las películas del cine. Para los más modestos, es suficiente con cumplir su cometido aunque no sea una gran producción, sin embargo hay que tener siempre en cuenta mantener, ante todo, la calidad aun siendo una sola pantalla.



En la imagen se muestra una animación de presentación realizada con el propio motor 3D del juego.

Si decidimos realizar una pequeña animación para presentar nuestro juego, hay varias formas de hacerlo. Se puede utilizar tanto el propio motor gráfico del juego como los modelos, con lo que perderíamos la espectacularidad que ofrece las animaciones creadas con paquetes 3D. Sin

embargo, esta técnica es ideal para largas presentaciones y mostrará al jugador el aspecto real que tendrá el juego. Por otro lado, la creación de una película de animación conlleva la utilización de aplicaciones 3D especializadas, lo que significa un mayor desembolso. Se consume más espacio en el disco y se deben utilizar técnicas de compresión de vídeo para que su reproducción sea posible en gran variedad de sistemas, lo que no siempre da el resultado deseado.

Si en nuestro diseño sólo contemplamos algunas pantallas fijas mostrando un texto con el argumento (como en el caso de "Zone of Fighters"), podemos añadir algún tipo de voz narrativa con música de fondo para crear más ambiente y en cierta manera obligar al jugador a leer todo el texto. Resumiendo, el argumento debe presentarse al usuario del juego de una forma atractiva, ordenada y con texto o voz claros y concisos.

(■) TUTORIALES DE AYUDA

Estos elementos son importantísimos en el desarrollo de un juego. Los tutoriales son el manual de usuario y resultan necesarios para transmitir el procedimiento a seguir para sacar el mayor rendimiento al juego. Evidentemente, dependiendo del juego, los tutoriales serán de una forma u otra. Juegos de una temática de uso compleja, como los de estrategia o rol, suelen poseer tutoriales interactivos muy completos, normalmente incluidos en el menú principal. Otros incluyen niveles de aprendizaje o entrenamiento y los más simples, algunos textos explicativos.

La forma más corriente de implementar un tutorial es de forma externa, incluido en la instalación del juego. Podemos encontrar tutoriales para leer con procesadores de texto o en formato HTML para cualquier visualizador web. En todos debe aparecer una descripción del juego y de todos los elementos que lo componen, una relación de todos los controles disponibles para el jugador y por último la lista de requerimientos técnicos mínimos necesarios para el funcionamiento del juego.

Una vez dentro del juego también debemos procurar una opción que muestre la lista de controles y la misión a cumplir. Es muy normal que el jugador no lea las instrucciones del juego, y sin perder tiempo empiece a jugar sin saber siquiera cómo se maneja su personaje. Por eso tenemos que tener esta situación prevista y proporcionarle en cualquier momento la posibilidad de acceder a una ayuda breve durante la partida. Puede valer algún esquema o dibujo o simplemente una ventana de texto en la pantalla.

En "Zone of Fighters" diseñaremos un tutorial en HTML de forma externa y un panel gráfico dentro del juego que mostrará una relación de controles. Además, implementaremos una tecla para que cuando el jugador la pulse pueda ver de nuevo la lista de controles en cualquier momento de la partida y sin ningún tipo de interrupción.



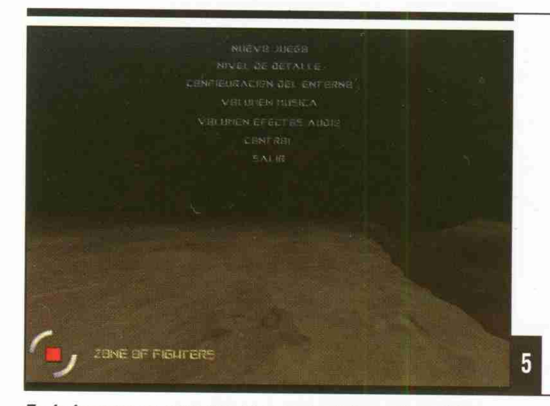
Es importante que el jugador pueda disponer durante la partida de un cuadro de ayuda con una descripción de los controles.

(■) LOS MENÚS

Con el menú entramos definitivamente en el juego. Está formado por numerosas opciones que a su vez conducen a otros submenús. El diseño de los menús es una parte importantísima y debe poseer la misma estética que el resto del juego, además de una interfaz sencilla de usar.

La configuración de las características del juego se realiza en este apartado y todo depende de la cantidad de opciones que le ofrecemos al usuario. Hay menús más o menos completos y con una

Los tutoriales en HTML son un formato válido para acompañar la documentación de un juego.



En la imagen se muestra el aspecto del menú de "Zone of Fighters".

estructura y capacidades que dependen mucho del tipo de juego. Pero, básicamente, podemos distinguir las siguientes partes en el menú de un juego: opciones gráficas, de sonido, de control y de juego.

(■) OPCIONES DE JUEGO

Dependiendo del tipo de juego podemos configurar multitud de aspectos. Lo más básico es, simplemente, empezar a jugar. Sin embargo, hay juegos que por sus características necesitan ampliar este apartado, desde la configuración del personaje (aspecto, nombre) o la dificultad a la elección del nivel de partida.

En "Zone of Fighters" en un principio se dará la posibilidad de elegir entre dos tipos distintos de terrenos, pero posteriormente podemos incluir la elección de mapas de combate creados con el editor de zonas o la generación aleatoria por parte del ordenador.

(■) OPCIONES GRÁFICAS

Debido a la gran cantidad de sistemas de diferente potencia gráfica, es necesario incorporar al juego una serie de opciones que permitan al usuario regular el rendimiento del juego. Principalmente se permite controlar la resolución de pantalla, la cantidad de colores, la elección de tarjeta gráfica u otros aspectos más técnicos como la configuración del entorno de juego como luces, sombras, etc.

En nuestro caso hemos preferido diseñar un menú aparte del menú principal del juego para elegir la resolución y potencia gráficas. Ya dentro del juego diferenciamos dos opciones, una para el nivel de detalle y otra para configurar el entorno. Esto es así dadas las características de "Zone of Fighters", del tipo de juego que es y de dónde se desarrolla. Al elegir "Nivel de detalle" entramos en otro submenú que nos permite seleccionar las siguientes opciones:

- **Detalle del terreno.** Bajo, medio y alto. Aquí elegimos la cantidad de polígonos que tendrá el terreno de juego. Por defecto colocamos el detalle en un nivel medio.
- **Sombreado del terreno.** Podemos activar o desactivar este efecto llamado *shading* y que produce sombra en el terreno dependiendo de la posición de la luz. Por defecto se encuentra activado.
- **Antialisado.** Esta opción permite activar o desactivar el suavizado de bordes de los polígonos. Inicialmente desactivado.
- **Rango de visión de la cámara.** Bajo, medio y alto. Con esta opción podemos ampliar hasta dónde es capaz de ver la cámara. Por defecto activamos un rango medio.



Las opciones de configuración gráficas deben ser todo lo variadas posible.

En el submenú "configuración del entorno", definimos todo el aspecto que tendrá la ambientación del juego. Diseñamos las siguientes:

- **Activación o no de niebla.** Por defecto está activada. En caso de quitarla del entorno debemos situar el rango de visualización más allá del límite de las paredes de la urna para evitar los cortes en la vista.
- **Activación o no del cielo y del reflejo del agua.** Por defecto están desactivados.
- **Activar el paso del día a la noche y viceversa.** Por defecto está activado y consiste en una disminución de la intensidad de la luz ambiental a la vez que se rota en su eje X.
- **Cantidad de decorado.** Bajo, medio, alto. Por defecto colocamos una cantidad media. Esta opción sólo tiene efecto cuando se elige la opción de juego de zona de combate aleatoria, y consiste en colocar una cantidad baja, media o alta de objetos de decorado sobre el terreno.
- **Activar o no el terreno dinámico.** Por defecto está activado y su finalidad es permitir la creación de agujeros producidos por los impactos de las bombas de minifusión.
- **Activar o desactivar decorado interactivo.** Por defecto está también activado y consiste en que los disparos dejen o no huella en los edificios en forma de manchas negras.

(■) OPCIONES DE SONIDO

En este apartado simplemente se permite el cambio de volúmenes del audio del juego, desde la música hasta los efectos sonoros como disparos, explosiones o voces.

Para nuestro juego, diseñamos la posibilidad de elegir el volumen de la música y de los efectos de audio en niveles bajo, medio y alto.

(■) OPCIONES DE CONTROL

Esta parte del menú es muy importante. Aquí se informa al jugador de cuáles son los controles para manejar el juego, además de permitir elegir entre diferentes dispositivos como teclado, ratón, joystick o *gamepad* (mando de juegos).

Es muy común permitir también el cambio por parte

del jugador de las teclas de control comúnmente llamado "configuración de teclas". "Zone of Fighters" tiene unos controles fijos, no pueden ser configurables, así que mostraremos un panel gráfico con la lista de controles. El juego sería capaz de jugarse perfectamente sólo con el ratón, aunque habrá más controles para manejar las vistas de cámara y la activación del escudo y camuflaje.

(■) ASPECTO DE LA PANTALLA DE JUEGO. INDICADORES

Los indicadores juegan un papel importante, ya que su misión es informar al jugador de todo lo concerniente al desarrollo de la partida y al estado de su personaje. También se incorporan otros elementos como sistemas de guía o ayuda.

Pueden ser todo lo variados que la imaginación pueda alcanzar y su aspecto tiene que ir en concordancia con el juego. Lo fundamental es que cumplan su misión de manera clara y concisa ya que multitud de ellos en la pantalla pueden incluso molestar al desarrollo de la partida.

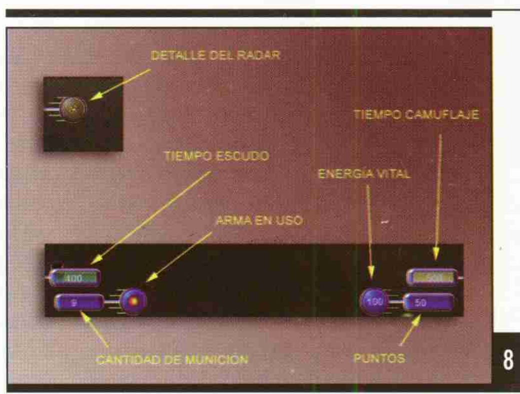
Vamos a diseñar para nuestro juego unos indicadores más o menos estándar y muy claros de distinguir. La idea es colocarlos en la pantalla de forma estratégica para que la vista de la acción sea lo más grande posible. Podríamos situar sólo números sobre la pantalla, pero seguramente pasarían desapercibidos y con el grueso de la acción serían prácticamente invisibles al jugador. Por eso, vamos a impresionar estas cantidades sobre algún tipo de gráfico que simule una pequeña pantalla de cristal. Podemos jugar con los colores y situación de estas pantallas para que el jugador las identifique rápidamente sin tener que desviar mucho la mirada.

Utilizaremos para estos indicadores un diseño geométrico y estilizado, sin muchos adornos. Estudiando las características del juego tendremos que informar al jugador con siete indicadores distintos situados a ambos lados de la pantalla en grupo de tres. El séptimo indicador será un radar, que muestre por medio de puntos de colores la situación del jugador (punto azul), de los demás participantes (puntos rojos) y de los voladores (puntos amarillos). Estará situada discretamente en la esquina superior izquierda de la pantalla y con forma esférica.

Colocamos en la parte inferior derecha el grupo de tres indicadores para la energía vital de la bionave, la cantidad de puntos conseguidos y el tiempo restante de camuflaje. En la parte inferior izquierda encontraremos un indicador con un gráfico del arma en uso y otro con la cantidad de munición de dicha arma y por último la cantidad restante de tiempo de escudo.

Otros tipos de indicadores son necesarios para "Zone of Fighters", como el punto de mira para la vista en primera persona y una cantidad numérica en la parte central superior de la pantalla que aparece cuando algún luchador, planta o animal es alcanzado por un disparo, indicando su nivel vital. Para el punto de mira se han diseñado dos tipos diferentes dependiendo del arma que se elija: uno alargado para la munición de bajo calibre y misiles (ya que la bionave dispara por cañones situados a los lados) y otro circular para el disparo de bombas.

Hasta aquí tenemos ya diseñado todos los aspectos que forman "Zone of Fighters". Podemos aplicar, fácilmente, todas las ideas expuestas a otros juegos y lo importante es saber que cuando se diseña un juego hay que pensar siempre en el jugador. Conseguiremos de esta forma que nuestro juego guste y sea aceptado por la comunidad.



Una descripción de los indicadores de pantalla del juego "Zone of Fighters".

7

Un ejemplo de diseño de indicadores de pantalla.



En el próximo número...

... empezaremos la programación de "Zone of Fighters", pero antes debemos prepararnos algunas herramientas que faciliten el trabajo de desarrollo, como visualizadores de modelos y terrenos, y plantearnos un editor de zonas de combate.

Grafismo 2D. La Tipografía

En el número anterior completamos la realización del logotipo y la pantalla de presentación de nuestro juego. Realizaremos en esta entrega nuestro propio tipo de letra, modificando una fuente compatible con Windows y, seguidamente, hablaremos un poco sobre los formatos 3D que utilizaremos en "Zone of Fighters".

Para nuestra fuente elegiremos de base el tipo "dungeon" que ya utilizábamos en la pantalla de presentación y cambiaremos el perfil de las letras desde la "A" hasta la "Z" mayúsculas para adaptarlas a nuestro diseño con el programa Font Creator.

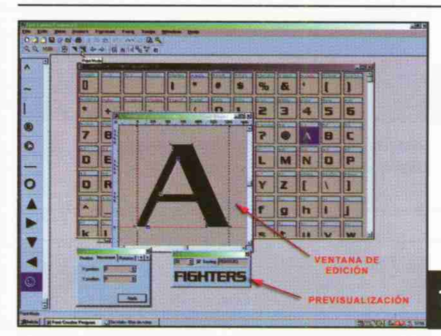
LA TIPOGRAFÍA

Los tipos de letras son muy importantes para reconocer nuestros productos y darles personalidad. Por ejemplo, en publicidad es conocido el poder que tienen para identificar una determinada marca. Por ello, se debe prestar atención en su diseño como una parte gráfica más del juego. Para construir un tipo de letra propio se pueden utilizar varias técnicas, como son dibujarlas o crear una fuente compatible con Windows. Dibujar un alfabeto, convirtiendo cada letra en un *sprite*, ha sido la manera tradicional de realizar una tipografía. Con este sistema es fácil generar una letra colorista y de formas muy variadas obteniendo resultados realmente originales. El inconveniente radica en su utilización dentro del juego ya que, al tratarse de imágenes, no resulta tan fácil imprimir un texto como lo haríamos utilizando una fuente *true type font* para Windows. Tendríamos que sustituir los valores ASCII por su correspondiente en imagen. Realizaremos un estudio de cómo funciona este procedimiento en uno de los tutoriales de programación de próximos números. Por el momento, en "Zone of Fighters" utilizaremos una tipografía compatible con Windows, ya que es un método más fácil de implementar y nos servirá para otros tipos de grafismos; por ejemplo, sustituyendo la letra del título del juego por la nueva que hayamos creado.

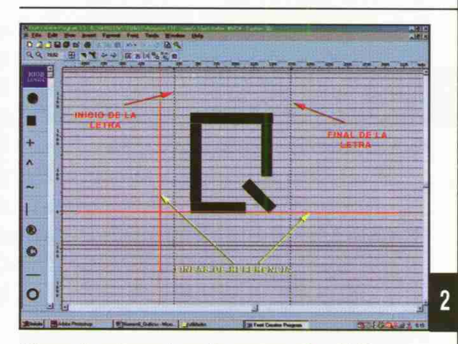
CREANDO NUESTRA PROPIA FUENTE

Una vez dentro del programa *Font Creator* tenemos que abrir la fuente *dungeon.ttf*. Para ello elegimos la opción "Installed Font" en "Open" del menú "File" o pulsando CTRL + I. Aparecerá una lista de tipos de letras. Elegimos la "dungeon normal (true type)". Una vez cargada, se mostrarán en una ventana todos los caracteres de la fuente con la tecla asociada a cada uno de ellos. Podemos apreciar también en la parte inferior una ventana de diálogo llamada "Preview". En ella veremos cómo van quedando nuestras letras, escribiendo en la línea de texto y eligiendo un tamaño. Por el momento, vamos a escribir "ABCDEF" con mayúsculas y un tamaño de 30. A medida que modifiquemos estas letras, veremos los cambios efectuados inmediatamente en esta ventana. Empezamos con la letra "A" mayúscula. Si pulsamos dos veces seguidas sobre ella activaremos la ventana de edición (Fig. 1).

Nos encontramos con la letra sobre una cuadrícula, delimitada por dos líneas discontinuas que representan el comienzo y el final global de la letra y dos ejes de color rojo (líneas de referencia) que indican la posición vertical y horizontal con respecto a la línea



Al pulsar dos veces sobre la casilla de la letra abriremos la ventana de edición.

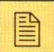


Líneas de referencia en la ventana de edición.

DEFINICIÓN

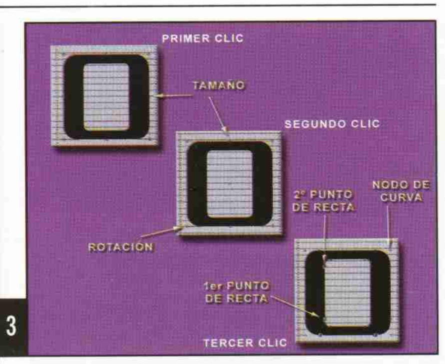
▸ TRUE TYPE FONT

True Type Font **es simplemente un tipo de letra o fuente usado por todas las aplicaciones que corren bajo Windows. El archivo tiene la extensión .TTF y almacena una definición de los caracteres de dicha fuente.**



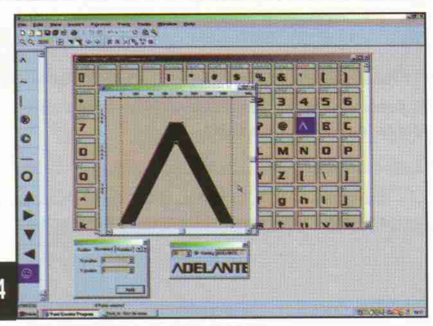
NOTA

Podemos modificar el aspecto, tamaño, etc. del grid o cuadrícula en "Tools/Grid Options", así como el color y la forma de las líneas de referencia.



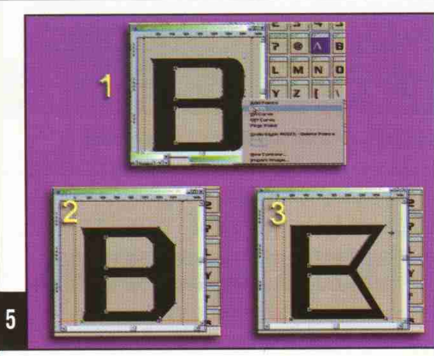
3

Distintos modos de edición según el número de clics que hagamos con el ratón.



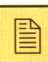
4

Aspecto definitivo de la letra "A".



5

Procesos para formar la letra "B".



NOTA

Es importante saber que moviendo las líneas discontinuas que marcan el principio y final de la letra, podemos crear más o menos espacio entre letras dentro de una palabra.

de texto. Sin dejar de pulsar sobre la letra y desplazando el ratón, la moveremos de posición en la línea de texto (Fig. 2).

Observamos que, al pulsar sobre ésta, se rodea de un cuadrado con seis cuadraditos o nodos, que nos servirán para modificar su tamaño. Si pulsamos dos veces, los nodos cuadrados se vuelven redondos y rectangulares; éstos nos servirán para transformarla y rotarla, y si pulsamos tres veces entraremos en la edición por puntos. Pulsando fuera de la letra volveremos al estado original de modo de contorno (Fig. 3).

Empezamos a modificar la letra "A" activando el "Point mode" o edición por puntos, pulsando tres veces sobre ella o activando la opción "View/Mode/Point". Los puntos de edición cuadrados representan comienzos y finales de líneas rectas y los puntos redondos nodos de control de curvas. Seleccionamos estos puntos redondos y los borramos. Pulsamos sobre cada uno de ellos con el botón derecho y elegimos "delete" en el menú emergente o simplemente pulsamos la tecla de suprimir. A continuación, desplazamos los puntos de rectas hasta formar la figura que se muestra en la imagen 4.

Pulsamos el icono de "Next Glyph" para pasar a la siguiente letra. Si en vez de hacer esto pulsamos dos veces sobre la siguiente letra, abriremos otra ventana de edición. Para la "B" realizaremos la misma operación que con la "A" con los nodos de curvas, es decir, los borramos. Podemos seleccionar más de un punto a la vez remarcándolo con el ratón o seleccionando uno a uno con la tecla SHIFT pulsada (Fig. 5).

Seguidamente, movemos los puntos de control cuadrados hasta obtener la forma que se muestra en el último cuadro de la figura 5. Para la letra "C" vamos a borrar la parte superior para quedarnos con forma de "L". Seleccionamos con el ratón todos los puntos de la parte superior de la letra y borramos. Posteriormente, eliminamos también todos los nodos de control de curvas. Para la parte superior, desplazaremos la barra horizontal de la librería, situada a la izquierda del programa, hacia la ventana de edición y le aumentaremos el ancho (Fig. 6).

La siguiente letra, la "D", es muy parecida a la "C", pero con una pequeña modificación; así que podemos copiar la "C" en el lugar de la "D" y modificarla. Pulsamos con el botón derecho sobre la casilla de la "C" y elegimos "Copy" (CTRL + C). Pulsamos sobre la letra "D" de igual forma y elegimos "Paste" (CTRL + V) (Fig. 8).

Luego seleccionamos la barra vertical de la letra hacia la derecha hasta obtener la forma de una "D" (Fig. 9).

La "E" es exactamente igual que la "C" pero con una barra horizontal en el centro. Entonces, podemos copiar la "C" en la casilla de la "E", pulsamos sobre la barra horizontal superior y hacemos CTRL + C y CTRL + V para hacer una copia de ella. Hay que tener en cuenta que la copia se realiza sobre el original; si pulsamos sobre ésta y desplazamos hacia abajo, veremos cómo movemos la copia. La "F" es igual que la "E" sin la barra horizontal inferior, y así sucesivamente con todas las demás letras. Las que son parecidas se copian y luego se transforman hasta obtener el resultado deseado. En la figura 10 se muestra la lista de todas las letras nuevas para "Zone of Fighters", desde la "A" hasta la "Z" mayúsculas, para que sirva de guía. A continuación, mostramos cómo realizar algunas operaciones importantes en el modo *edición por puntos*:

- **A.** Los puntos cuadrados de color verde son los primeros de su línea y los azules los consecutivos. Para convertir los puntos consecutivos en primeros y los que le siguen en secundarios hay que pulsar el botón derecho sobre el punto azul o secundario que queremos convertir y elegir la opción "First Point".

- **C.** Para crear un nuevo punto de edición en una línea, pulsamos en cualquiera de los puntos de dicha línea con el botón derecho y elegimos la opción "Add Point". Se creará un punto entre éste y el anterior, a no ser que realice sobre un punto primario (First Point), en cuyo caso el nuevo punto aparecerá a continuación.
- **D.** Para convertir un punto de línea (punto cuadrado) en curva, pulsamos con el botón derecho y elegimos la opción "Off Curve" en el menú emergente; para la operación contraria, en "On Curve".

Una vez que tenemos nuestra fuente terminada, podemos comprobar los resultados abriendo la ventana "Test Font" pulsando la tecla "F5". Antes de instalarla debemos guardarla con otro nombre, por ejemplo, "ZOF.TTF" con la opción "File/Save as" en el directorio "C:\Windows\Fonts". Para instalar la fuente elegimos la opción "Install" del menú "Fonts" (Fig. 11).

FORMATOS DE OBJETOS 3D

En el mundo de las aplicaciones 3D existen multitud de formatos de objetos diferentes. Sin embargo, todos tienen en común el ser utilizados para almacenar objetos y relacionarlos entre sí. Prácticamente cada programa tiene su propio formato. Aun así, se han mantenido de uso común una serie de ellos que por su popularidad han creado estándares. Generalmente, este hecho se ha producido por el establecimiento en el mercado durante los comienzos del 3D de aquellos programas más usados. Es el caso de 3D Studio de Autodesk que implantó su formato .3DS en todo el mundo y hoy día es, con toda seguridad, el más usado en el desarrollo 3D, generalmente para el intercambio de objetos entre aplicaciones.

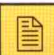
.3DS

Es un formato sencillo de implementar en motores 3D debido a su estructura por trozos de datos almacenados jerárquicamente. Para el diseño de videojuegos es muy interesante porque, además de estar soportado por todos los programas de diseño 3D, puede llevar asignadas las texturas y la animación en el mismo fichero.

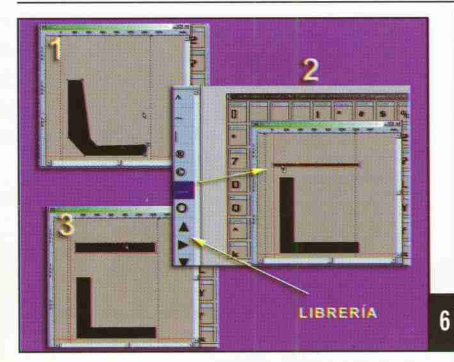
Blitz3D soporta este formato y aprovecharemos esta ventaja para utilizarlo en los objetos estáticos que aparecen en "Zone of Fighters": edificaciones, rocas, árboles y en la planta "Shaark". Aunque también podemos utilizar objetos con animación en .3DS, ya que Blitz3D es capaz de obtener la animación y la jerarquía almacenados. Si queremos utilizar la opción de colisión por polígonos del Blitz3D, resulta conveniente el uso de este formato en objetos animados para poder obtener unos resultados óptimos. Además, si en nuestra aplicación 3D asignamos las texturas al objeto y las incluimos en el mismo directorio que éste, no es necesario aplicarlas desde programación. Simplemente con cargar el objeto es suficiente para que se dibuje texturizado.

XOF O .X

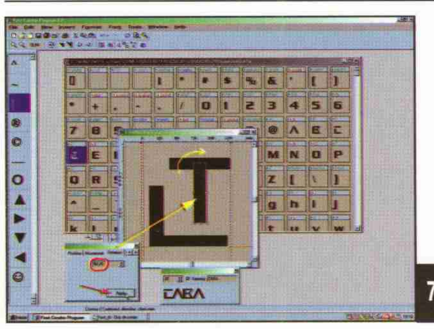
Hay un formato estándar en prácticamente todas las aplicaciones multimedia programadas con la API de Microsoft DirectX, nos referimos al XoF o .X. Aunque este formato es el estándar de DirectX, no es soportado por la mayoría de los programas 3D. Comúnmente se trabaja con formatos como .3DS y luego son convertidos, con aplicaciones especiales, a .X. Lo que resulta atractivo en este formato es su utilización en motores gráficos, ya que se compone de una estructura en forma de plantilla que el programador puede utilizar y modificar a su gusto o bien usar plantillas base. En este formato también se utilizan árboles jerárquicos e igualmente almacena objetos, materiales y los keyframes de animación.



Blitz3D soporta este tipo de objetos y se comporta de la misma forma que el .3DS. Podemos exportar a este tipo de formato con la aplicación de modelado en



6 *Procesos para formar la letra "L".*



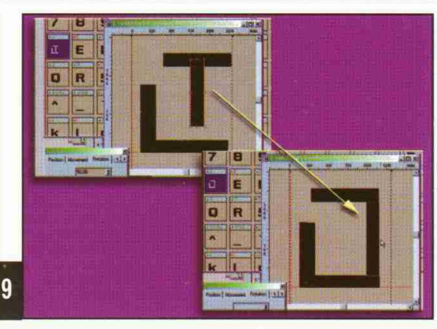
7 *Procedimiento para rotar una porción de la letra.*

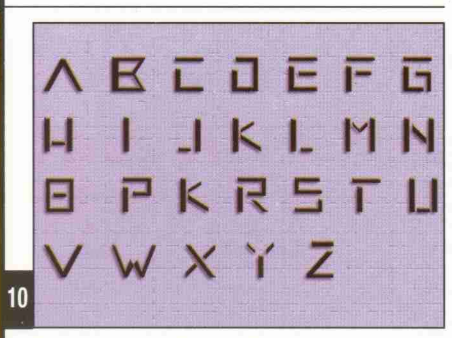


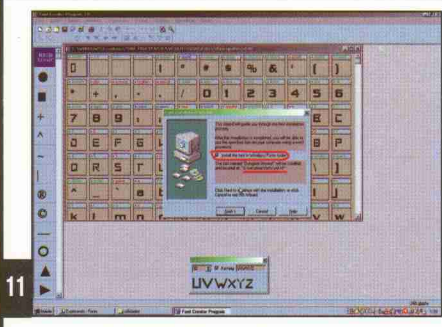
8 *Podemos copiar una casilla a otra para duplicar letras.*

TRUCO

Hay otra manera de realizar las operaciones de movimiento, transformación, rotación, etc. de una forma precisa numéricamente en *Font Creator*, y es por medio de la ventana de diálogo "Transform", la cual se activa en "View/Toolbars/Transformations" (Fig. 7).

9

Con pocas modificaciones se puede obtener una letra nueva a partir de otra.

10

Set de letras para Zone of Fighters.

11

Ventana de instalación para la nueva fuente.


TRUCO

Antes de empezar a crear tu propia fuente, fíjate bien en qué letras son parecidas entre sí para, de este modo, no trabajar dos veces. Podremos usar una misma letra como plantilla, y a partir de ésta formar todas las que se le asemejen.

baja Milkshape3D e importarlos desde el mapeador UV Lithunwrap , que utilizaremos en esta obra.

.OBJ

Este formato es un estándar de *Alias WaveFront Object* y es muy común su uso para pasar la geometría de los objetos de una aplicación 3D a otra. También es muy utilizado en visualizadores de objetos 3D. Este formato soporta tanto objetos poligonales (puntos, líneas y caras) como objetos de forma libre (curvas y superficies) y además puede ser almacenado en formato ASCII y binario, aunque el más utilizado es el primero de ellos.

Blitz3D no puede leer este tipo de ficheros, pero es el utilizado (además del .3DS) por *Character FX*, que utilizaremos para animar nuestros personajes. Así que tendremos que exportar a .OBJ o .3DS desde el modelador, para poder trabajar con los objetos desde *Character FX*, aunque en esta obra importaremos los objetos desde esta animación en formato .OBJ como un solo mesh.

.LWO

Este formato es el utilizado por la popular aplicación *3D LightWave*. Es un estupendo formato para almacenar animación, además, claro está, de objetos poligonales, curvas, superficies y texturas. Cada vez es más soportado por la mayoría de las aplicaciones más populares.

En nuestra obra no utilizaremos este formato porque no puede ser leído por Blitz3D directamente.

.MD2

Por último, este formato es una versión avanzada del .MDL impuesto por la empresa ID Software para su juego Quake. Este tipo de archivo contiene la geometría del modelo, información sobre la animación y las coordenadas y nombres de las texturas.

Se caracteriza por dar lugar a animaciones suaves gracias a su sistema de deformación de la malla que forma el objeto.

Este formato sí es soportado por Blitz3D y es una opción muy válida para utilizar objetos animados en nuestros juegos. El único inconveniente que presenta, es que la detección de un objeto animado en este formato no es muy precisa y sólo funciona con el tipo de colisión esfera a esfera. Este formato es el que utiliza *Character FX* para exportar las animaciones. Nos encontramos entonces con un dilema. Si utilizamos *Character FX* para nuestros propósitos, no tendremos más remedio que utilizar este formato en nuestro juego, con los problemas que conlleva. Sin embargo, la versión 1.76 de Blitz3D incorpora la utilización de un formato propio, el .B3D

.B3D

Este formato fue originando por el equipo creador del Blitz3D para la versión 1.76 de este lenguaje de programación. Su principal característica es que engloba geometría de objetos, texturas y frames de animación. Es decir, tiene la animación del formato .MD2 y las características geométricas de los formatos .3DS o .X.

Así que podemos poseer un objeto con animación por deformación de malla que puede ser detectado con todo tipo de colisiones (polígonos, cúbica, esférica).

Éste es el formato que utilizaremos en nuestro juego para los personajes animados como las plantas o animales. Para conseguir exportar un objeto .3DS o .OBJ al formato .B3D desde *Milkshape3D* o al mismísimo *Character FX*, están disponibles plugins para poder realizar esta operación en estas dos aplicaciones. Además podemos disponer también de un plugin para exportar a .B3D desde el propio *LightWave3D*.


En el próximo número...

... modelaremos nuestra bionave de combate con Milkshape3D.

Cómo desarrollar los efectos especiales de sonido (II)

Después de conocer cómo configurar Goldwave para la grabación y reproducción de audio, es el momento de entrar realmente en materia. Vamos a crear nuestro primer efecto de sonido partiendo de cero. Con él aprenderemos la aplicación de filtros y efectos de audio para obtener diferentes resultados con un rendimiento en tamaño y calidad aceptables.

NORMALIZAR Y ECUALIZAR

Cuando se graba un sonido desde una fuente exterior, no siempre se obtiene el aspecto sonoro deseado. En algunas ocasiones, suena demasiado agudo o demasiado grave, y en otras, con volumen muy bajo.

Podemos hacer una primera aproximación a la calidad que deseamos por medio de un par de funciones muy importantes: el normalizado y la ecualización.

Por medio de la normalización, podemos ajustar el volumen del sonido al límite máximo, antes de sufrir saturación. Por otro lado, ecualizando podemos cambiarle los rangos de frecuencia y variar de esta forma su tonalidad. Conseguimos así un sonido más rico y con cuerpo.

Cojamos una bolsa de plástico. Vamos a crear el sonido de un fuego. Será perfecto para comprender mejor el normalizado y la ecualización. Si manoseamos la bolsa con cuidado, notaremos que el sonido que produce se asemeja al del fuego quemando leña.

Preparamos el editor para grabar. Ajustamos el volumen del micrófono y deseleccionamos la opción "Loop" y "Ctrl key safety" de las propiedades de grabación. Creamos una plantilla vacía de 10 segundos de duración con calidad CD y grabamos.

Observamos que la grabación suena algo baja de volumen. Pues bien, vamos a ajustar su volumen al máximo sin saturar. En Goldwave tenemos varias opciones para amplificar el volumen en "Volume", en el menú "Effect". Para normalizar debemos seleccionar "Maximize", pinchar la opción "New maximum" y colocar el valor en 1.0. La opción "New RMS" es útil para igualar el volumen de varios ficheros de audio distintos, por ejemplo, de varias músicas para un CD (Fig. 1).

Una vez ajustado el volumen, vamos a darle más cuerpo al sonido por medio de la ecualización. Elegimos "Filter/Equalizer" en el menú "Effects". Entramos en un pequeño equalizador de siete bandas con frecuencias de 60 Hz (sonido grave) hasta 15 Khz



DEFINICIÓN

▶ DECIBELIOS

Los decibelios son una escala logarítmica de intensidad de la señal de audio; es decir, miden el volumen de un sonido.



NOTA

Siempre es conveniente grabar y editar con la máxima calidad cualquier sonido, para posteriormente pasar a la resolución deseada. Con este sistema, conseguiremos unos mejores resultados.



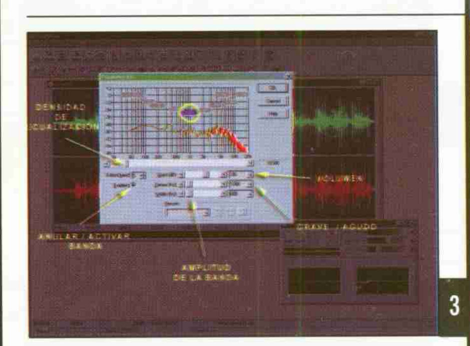
1

Mediante la función "Maximize volume" podemos obtener la normalización de nuestro sonido.



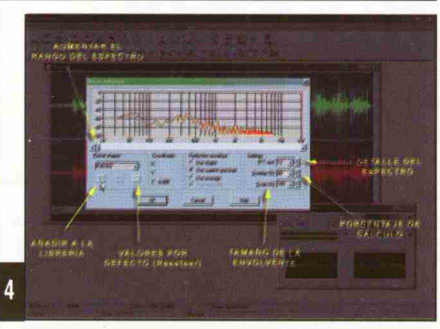
2

Descripción básica del ecualizado.

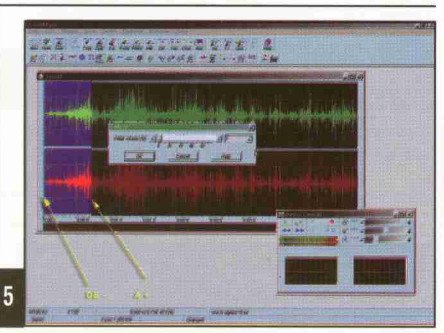


3

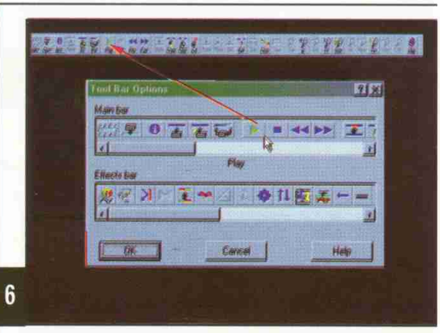
Descripción de la ventana del ecualizador paramétrico. (Imagen incluida en el CD-ROM).


4

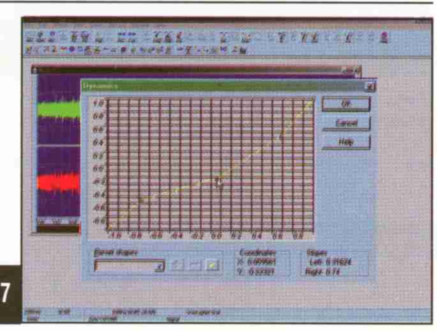
Descripción de la ventana del proceso de reducción de ruido.


5

Mediante la función "Fade in" podemos realizar una subida de volumen.


6

A través de la opción "Options / Tool Bar" puedes elegir qué iconos se mostrarán o cuáles no, en la barra de herramientas y de efectos.


7

Mediante el efecto "Effects /Dynamics" podemos alterar el mapeado de amplitud de una selección a lo largo del tiempo.

(sonido agudo). El volumen de cada una de estas frecuencias (bandas) va desde -20 decibelios hasta 10 Db. Por ejemplo, si aumentamos a 10 Db la banda de 15 Khz, estamos subiendo al máximo la frecuencia más aguda del sonido (Fig. 2).

Un sonido con demasiados graves puede sonar sucio y sin claridad. Si, por el contrario, tiene demasiados agudos puede aparecer ruido. Disponemos de otro ecualizador mucho más flexible, potente y preciso: el paramétrico. Lo podemos encontrar en "Effects/Filter/Parametric EQ". Su función es la misma, la diferencia es un mayor control sobre las bandas y el rango de frecuencias y volúmenes que podemos configurar repartidos en 30 bandas (Fig. 3).

APLICACIÓN DE EFECTOS

Aplicando todo lo que hemos aprendido, vamos a crear un disparo a partir de una palmada. Prepara el editor para grabar. Crea una plantilla nueva de 5 segundos de duración con calidad CD. Empieza a grabar dando una palmada seca a unos 20 centímetros del micrófono para no saturar la grabación.

Una vez realizada la grabación, debemos eliminar el espacio vacío del comienzo y la coletilla de audio inservible del final.

NOTA

Recuerda que la marca de principio de selección se mueve con el botón izquierdo del ratón, y la de final de selección, con el botón derecho.

DEFINICIÓN

▸ ENVOLVENTE

La envolvente representa la dinámica de un sonido, es decir, los cambios que sufre en la frecuencia y el volumen.

Los señalamos con las marcas de posición de selección y borramos con CTRL + X. Aunque la grabación parezca tener un buen volumen, siempre es conveniente normalizarla.

Una vez normalizado el volumen, podemos quitar un poco de ruido con un filtro de reducción de ruido "Noise reduction", situado en "Effects/Filtres". Podemos descubrir una gráfica del espectro del sonido con la frecuencia en el eje X y los decibelios en el Y, además de una serie de opciones. Modificando las opciones en "Settings", podemos conseguir buenos resultados ya que determina las características del espectro. El detalle de éste lo determina la opción "FFT size", así que lo situamos en 12. La opción "Overlap" determina el porcentaje de cálculo del espectro. Le asignaremos un valor del 80%. Y con "Scale" podemos modificar el tamaño de la envolvente del sonido, con lo que a valores altos borramos más audio del sonido, así que los situamos en 100 % (Fig. 4).

A continuación, vamos a reducir el tono para obtener un sonido más alargado y grave. Elegimos la opción "Pitch" en "Effects". Deseleccionamos la opción "Preserve lenght" para no mantener la misma longitud. Activamos "Semitone" y elegimos el valor -12. Realizamos la operación dos veces, es decir, un semitono a -24. Para reducir el tiempo de ataque del comienzo, vamos a aplicar un efecto de menos a más volumen llamado "Fade in". Seleccionamos desde el principio hasta un tercio del sonido y elegimos la opción "Effects/Volume/Fade in" (Fig. 5).

Haremos lo mismo, pero con la operación contraria, seleccionando el final del sonido y aplicando "Effects/Volume/Fade out". Podemos aplicar cuantos efectos especiales nos apetezca hasta obtener el sonido que deseemos, por ejemplo, un poco de eco o un efecto *flanger*.

Manejo de texto e imágenes con Blitz 3D

El manejo de texto e imágenes es fundamental para el desarrollo de cualquier juego o aplicación multimedia.

Por ejemplo, las aventuras gráficas o las conversacionales se basan en estas dos cualidades. Aprenderemos a imprimir en pantalla y a controlar cadenas de texto, así como a cargar y controlar ficheros de imágenes.

IMPRIMIR TEXTO EN PANTALLA

La instrucción más básica para imprimir texto en pantalla es "Print":

```
Print "Aquí escribimos un texto"
```

Coloca el cursor en la posición (0,0) e imprime el *string* y luego pasa el cursor a una nueva línea. Si no queremos mover el cursor de línea debemos utilizar "Write":

```
Write "Aquí escribimos un texto"
```

Puede ocurrir que nos interese imprimir en cualquier parte de la pantalla. Para ello podemos utilizar la instrucción "Locate", la cual coloca el cursor en las coordenadas deseadas:

```
Locate Coordenada Horizontal,
Coordenada Vertical
Locate 10,20
Print "HOLA"
```



NOTA

Debemos saber que las instrucciones "Print" y "Write" escriben el texto en el búfer frontal ("FrontBuffer").

Existe, sin embargo, una función que engloba a las anteriores, se trata de "Text", que coloca y escribe un *string* en la pantalla gráfica:

```
Text Coordenada Horizontal,
coord. Vertical,"Texto",
centrado sX (true/false),
Centrado Y (true/false)
Text 200,100,"Aquí escribimos
un texto", true, falsee
```

Las dos últimas opciones se utilizan para centrar horizontal o verticalmente el texto.

Tanto a "Print", "Write" o "Text" podemos añadirles texto con el símbolo "+":

```
nombre$="Darío "
apellido$="Medina "
edad=1
Locate 100,10: Print nombre$+
apellido$+"tiene "+edad+" año"
Text 100,30, nombre$+
apellido$+"tiene "+edad+" año"
```

Resultado de ambos: Darío Medina tiene 1 año (Ver "ejemplo1.bb").

UTILIZANDO FUENTES DE LETRAS

En Blitz3D tenemos acceso a los tipos de fuentes instalados en Windows a excepción de las fuentes de símbolos como "Webdings". Para cargar una fuente utilizamos la función "LoadFont":

```
LoadFont ("nombre de la fuente",
altura true/false, negrita
true/false , itálica true/false,
subrayado true/False).
```

Y para poderla usar debemos activarla con *SetFont variable,* que contiene la fuente. Una vez usada la fuente se puede eliminar de memoria con la función:

```
FreeFont variable que contiene
la fuente.
```

En el "ejemplo1.bb" (CD-ROM, apartado "extras") podemos observar el funcionamiento de estas funciones.

MANEJANDO CADENAS DE TEXTO

Para empezar, comentaremos algunas pequeñas funciones útiles que siempre es bueno saber para optimizar mejor nuestro código.

Convertiremos nuestro texto a mayúsculas y a minúsculas con las instrucciones "Upper" y "Lower":

```
nombre$="Darío"
Print Upper(nombre$)
```

Resultado: DARÍO

```
nombre$="DArío"
Print Lower(nombre$)
```

Resultado: darío

Eliminamos los espacios posibles al principio y final de una frase con "Trim":

```
nombre$="    Darío     "
apellido$=" Medina "
Print Trim(nombre$)+ apellido$
```

Resultado: Darío Medina

Si queremos cambiar el tamaño de un *string*, podemos utilizar " Lset " y " Rset". Generalmente se utilizan para justificar el texto a la izquierda o a la derecha.

```
frase$="Mi coche verde "
frase$=Lset(frase$, 20)
Print ">"+frase$+"<"
```

Resultado:
>Mi coche verde <
"Mi coche verde" tiene 15 caracteres + 5 espacios = 20

```
frase$="Mi coche verde "
frase$=Lset(frase$,5)
Print ">"+frase$+"<"
```

Resultado: >Mi co<
"Mi coche verde" tiene 15 caracteres , sólo cogemos 5.

"Rset" funciona igual pero empieza a contar desde la derecha:

```
frase$="Mi coche verde "
frase$=Rset(frase$,20)
Print ">"+frase$+"<"
```

Resultado:
> Mi coche verde<

La instrucción "Replace" nos permite reemplazar los caracteres de un string por otro. Por ejemplo, si en la frase "Guadalajara tiene 6 a" queremos cambiar las "a" por "e", escribiríamos:

```
frase$="Guadalajara tiene 6 a"
Print frase$
Print Replace(frase$, "a", "b")
```

Resultado: Guedelejere tiene 6 e

Para encontrar la posición de un carácter o varios dentro de una frase se utiliza la instrucción "Instr":

```
Posicion = Instr (caracteres
a buscar, frase dóonde buscar,
posición de comienzo de búsqueda)
frase$="Guadalajara"
Print "La posición es: " +
Instr ( "a","frase$, 4)
```



DEFINICIÓN

▸ CÓDIGO ASCII

El código ASCII es un convenio que se adoptó para asignar un valor numérico a cada carácter.

Resultado: La posición es: 5

Empezamos a buscar la "a" a partir del cuarto carácter de la frase (la "d"), y obtendremos la posición de la primera "a" que encuentre a partir de ahí. Si no encuentra nada, devolverá el valor 0.

Para contar el número de caracteres de una frase (espacios incluidos) podemos utilizar la función "Len":

```
frase$="  hola"
Print Len (frase$)
```

Resultado:
6 (dos espacios y hola)

Para encontrar caracteres dentro de una frase disponemos de tres instrucciones "Left ", "Right" y "Mid". Por ejemplo, queremos obtener la palabra "Mi" y la palabra "verde" de la frase "Mi coche verde":

```
frase$ = "Mi coche verde"
palabra1$ = Left (frase$,2)
palabra2$ = Right (frase$,5)
Print frase$ : Print palabra1$
+" "+ palabra2$
```

Resultado: Mi coche verde
Mi Verde

Obtenemos los 2 caracteres más a la izquierda ("Mi") de la frase y los 5 más a la derecha ("verde")

Imaginemos que deseamos finalizar nuestro programa cuando el usuario introduzca el símbolo "*" en el ordenador.

La instrucción "Mid" nos puede ayudar en ello, ya que recorre un *string* en busca de uno o varios caracteres determinados.

```
For n = 1 to Len(entrada_usuario$) ;
recorremos la frase escrita por
el usuario en busca del asterisco
  If Mid(entrada_usuario$,"*"n,1)=
"*" then End
Next
```

Si queremos detectar por ejemplo dos caracteres, por ejemplo, "no", lo sustituiríamos por:

```
If Mid(entrada_usuario$,n,2)=
"no" then End
```

Recorremos la frase introducida por el usuario. Si a partir de la posición "n" hay 2 caracteres que coinciden con "no" finalizamos el programa.

Obtenemos los 2 caracteres más a la izquierda ("Mi") de la frase y los 5 más a la derecha ("verde")

En el "ejemplo3.bb" incluido en el CD se muestra un ejemplo útil del uso de estas instrucciones. Observad el uso de la instrucción "Right" para aislar la extensión de los ficheros y "Mid" para detectar el "." que separa la extensión. No es necesario entender, por el momento, los comandos referentes al uso de ficheros. Dedicaremos un capítulo del curso especialmente a este tema.

Para terminar, una función que nos permitirá repetir un número determinado de veces un *string*: "String":

```
String (frase a repetir,
numero de veces que se repite)
```

FUNCIONES DE CONVERSIÓN

En ocasiones, necesitaremos realizar operaciones con la tabla ASCII. Por ejemplo, saber cuál es el código numérico de un determinado carácter o viceversa. Para obtener este tipo de información disponemos de "Asc" y "Chr".

Queremos saber el código ASCII de la letra "A": *Print Asc("A").*

O lo contrario, el carácter correspondiente al código ASCII 65: *Print Chr(65).*

Y por último, también podemos convertir un valor numérico en un *string* con la instrucción " Str ":

```
numero=25
numero2$=Str(numero)
Print numero + numero
Print numero2$ + numero2$
```

Resultado: 50
2525
(Ver "ejemplo4.bb")

FUNCIONES DE IMAGEN

En los juegos es muy común el uso de imágenes para ilustrar o crear fondos fijos o animados. Por ejemplo: la pantalla de presentación, el logotipo pequeño en la esquina del menú o los indicadores de juego. Podemos manipular imágenes más grandes que la pantalla o de tan sólo pocos píxeles. También transformarlas de tamaño, moverlas por la pantalla e incluso detectar colisiones entre ellas. En Blitz3D disponemos de numerosas funciones para realizar todas estas operaciones. Aunque en este apartado sólo veremos las más usadas, son suficientes para cualquier proyecto de videojuego.

FUNCIONES DE DIBUJO

Debemos saber que, al igual que dibujamos en la pantalla "FrontBuffer" o en el "BakBuffer", podemos hacerlo directamente sobre una imagen con la función "ImageBuffer".

Es decir, podemos asignar que una imagen creada o cargada desde disco sea el búfer donde se dibuje. De esta forma podemos cargar una imagen y hacerle modificaciones sin que puedan verse o dibujar sobre una imagen nueva en tiempo real. En el programa "ejemplo6_5.bb" podemos ver un ejemplo de este procedimiento. (Para probar este ejemplo es necesario copiar al disco duro "C:\" el archivo de imagen "imagen.bmp" contenida en el CD). En el programa "ejemplo6_1.bb" podemos ver un ejemplo de este procedimiento.

Para cargar una de nuestras imágenes guardadas en el disco debemos utilizar la instrucción "LoadImage". Podremos utilizarlas en multitud de formatos como: bmp, jpg , gif, pcx o png. El archivo cargado lo asignamos a una variable global, la cual contendrá la dirección de memoria donde está la imagen guardada después de cargarla.

Hay varias instrucciones para dibujar nuestra imagen en la pantalla. Podemos dibujarla completa o sólo una porción rectangular de ella.

Para dibujar la imagen completa utilizando la máscara de color se utiliza "DrawImage" y para dibujarla sin ningún color transparente, "DrawBlock":

```
Global imagen
; La variable que manipula la imagen
Graphics 800,600,16
imagen=LoadImage("foto.bmp")
; Cargamos la imagen

DrawImage imagen,10,10
; Dibujamos la imagen en las coordenadas X=10 e Y=10 de la pantalla

DrawBlock imagen,200,10
```

Para dibujar una porción rectangular de la imagen utilizando la máscara de color se utiliza "DrawImageRect" y para dibujarla sin ningún color transparente, "DrawBlockRect".

Los parámetros para ambas son los mismos:

```
DrawImageRect variable que manipula la imagen, posición X, posición Y,
posición X del trozo rectangular,
posición Y del trozo rectangular,
tamaño horizontal del trozo,
tamaño vertical del trozo
```

Podemos salvar una imagen al disco en formato .bmp con la instrucción "SaveImage":

```
img = SaveImage (variable que manipula la imagen, "fondo.bmp")
```

Si por el contrario, queremos liberar de la memoria una imagen que ya no necesitamos, utilizaremos

```
"FreeImage" : FreeImage imagen
```

TRANSFORMACIÓN DE IMÁGENES

A cualquier imagen que hayamos creado o cargado desde disco, se le puede cambiar el tamaño. Disponemos de dos funciones para esta operación "ScaleImage" y "ResizeImage ":

```
ScaleImage variable que contiene la imagen, escala horizontal, escala vertical
```

Con esta función escalamos la imagen por porcentajes, es decir, un valor en "escala horizontal" y "escala vertical"de 1.0 significa un tamaño del 100%. Un valor de 2,0 sería 200% y un valor 0.5 sería la mitad del tamaño 50%. Tenemos una imagen de 200x100 píxeles y queremos aumentarla el doble horizontalmente y a la mitad de altura:

```
ScaleImage imagen,
2.0,0.5
```



NOTA

Para Blitz3D, el color negro (*rgb(0,0,0)*) de las imágenes es considerado como transparente y no se dibujará. A esto se le denomina *máscara de color*. Si queremos cambiar la máscara de color estándar a otro color que no sea el negro, es decir, cambiar el color transparente, debemos utilizar "MaskImage" :

```
MaskImage variable que manipula la imagen, valor de rojo, valor de verde, valor de azul.
```



NOTA

Hay que tener en cuenta que el modo gráfico se tiene que definir antes de cualquier operación de carga de gráficos, si no éstos se borrarían de la memoria.

```
ResizeImage variable que
contiene la imagen, tamaño
horizontal, tamaño vertical
```

Por el contrario, con esta función aumentaremos o disminuiremos el tamaño horizontal o vertical de la imagen por cantidad de píxel. Es decir, tenemos una imagen de 300 X 200 píxeles y queremos aumentarla al doble de su tamaño:

```
ResizeImage imagen,600,400
```

También podemos rotar una imagen cualquiera con la función " RotateImage ":

```
RotateImage variable que
contiene la imagen, valor
flotante en grados desde
0.00 hasta 360.00
```

Esta función es demasiado lenta para utilizarla en tiempo real, así que lo mejor sería realizar la rotación en el búfer oculto y luego mostrarla en la pantalla. Blitz3D aplica automáticamente un proceso llamado *filtro bilinear* , el cual suaviza la imagen para evitar los dientes de sierra. Para anular la aplicación de este filtro podemoss utilizar la función " TFormFilter ":

```
TFormFilter True (1. Activado)/
False (0. Desactivado)
```

COLISIONES ENTRE IMÁGENES

Es necesario disponer de procedimientos para el manejo de sprites en el desarrollo de juegos 2D o para implementar otros elementos presentes en todo tipo de juegos como la activación de botones en los menús por medio del ratón. Nos referimos a los sistemas de colisiones entre imágenes.

Encontramos funciones extremadamente potentes para este fin: "ImagesCollide", "ImagesOverlap".

Estas funciones detectan dos imágenes a nivel de píxel ignorándose el color transparente. Para que funcione necesitamos decirle a la función las dos imágenes que queremos detectar y la posición en la pantalla de cada una de ellas. Las dos funcionan exactamente igual, a diferencia de que "ImagesOverlap" es más rápida porque sólo detecta si una imagen esta sobre la otra:

```
ImageOverlap
ImagesCollide (imagen1, posicion1
X, posicion1 Y, frames1, imagen2,
posicon2 X, posicon2 Y, frames2)
```

Frame1 y frame2 se refieren al fotograma de animación de la imagen, en caso de que tuviese. Por ejemplo, el sprite animado de tu personaje o el puntero del ratón con diferentes aspectos.

Puede ocurrir que necesitemos detectar sólo una porción de la imagen. Es muy común este hecho en las aventuras gráficas, donde el puntero del ratón tiene que interactuar con partes de la imagen del escenario. Para realizar esta operación podemos utilizar las funciones "ImagesRectCollide" y "ImagesRectOverlap". Su funcionamiento es igual que las anteriores instrucciones, a diferencia de que detecta una imagen en contacto con una porción de imagen.:

```
ImagesRectOverlap
ImagesRectCollide (imagen, posicionX
imagen, posicionY imagen,
posicionX de la porción,
posicionY de la porción,
tamaño horizontal de la porción,
tamaño vertical de la porción)
```

Disponemos además, de una función para detectar si dos porciones de la pantalla están en contacto. Se trata de " RectsOverlap ":

```
RectsOverlap ( posicionX
porcion1, posicionY porcion1,
tamaño horizontal porción 1,
tamaño vertical porción 1,
posicionX porcion 2, posicionY
porcion 2,
tamaño horizontal porción 2,
tamaño vertical porción 2)
```

Se puede observar el funcionamiento de estas funciones en los ejemplos "ejemplo6_2.bb" y "ejemplo6_3.bb" que encontraremos en el cd.

INFORMACIÓN DE IMÁGENES

Conocer el tamaño o la posición de una imagen es fundamental para implementar correctamente todas las funciones de transformación y colisión. Para ello, Blitz3D ofrece diferentes funciones que nos proporcionan esta información: "ImageWidth", "ImageHeight", "ImageXHandle", "ImageYHandle":

```
ImageWidth (variable que
contiene la imagen)
```

Proporciona el tamaño horizontal de la imagen en píxeles

```
ImageHeight (variable que
contiene la imagen)
```

Proporciona el tamaño vertical de la imagen en píxeles.

```
ImageXHandle (variable que
contiene la imagen)
```

Proporciona la posición horizontal (X) de la imagen.

```
ImageYHandle (variable que
contiene la imagen)
```

Proporciona la posición vertical (Y) de la imagen.



En el próximo número...

... estudiaremos las posibilidades 2D del Blitz3D. Conoceremos estas capacidades en un mundo 3D, empezando por el manejo de sprites.

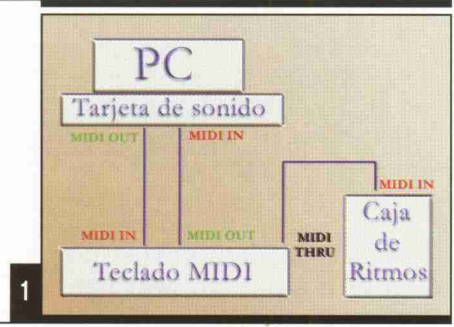
Secuenciadores. Cubase VST (III)

En esta entrega, terminaremos la serie básica de tutoriales que hemos dedicado a Cubase. Vamos a conocer el procedimiento a seguir para grabar música a través de la conexión MIDI, así como su correcta implementación en el conjunto de las pistas.

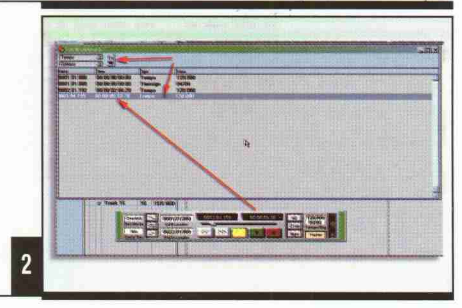
Para este tutorial, asumimos que hay un teclado MIDI conectado al ordenador por medio de la tarjeta de sonido como se muestra en la figura 1.

PREPARANDO LA GRABACIÓN MIDI

Antes de comenzar a grabar debemos ajustar las unidades de tiempo y ritmo en la barra de transporte o en la opción "Menu / Mastertrack / List", como se muestra en la figura 2.



Un esquema básico de conexión MIDI.

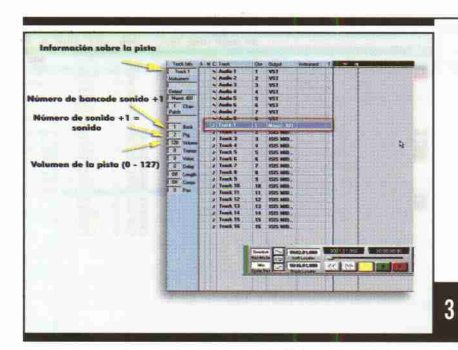


Procedimiento para insertar un nuevo tiempo y compás.

En la lista del Mastertrack podemos sumar todos los cambios de tiempo y de compás que queramos con solo pulsar el botón "In" después de seleccionar el punto de la canción en la barra de transporte.
Aconsejamos un tiempo de 120 y un ritmo 4/4. Activamos el clic del metrónomo. Elegimos la pista donde irá la grabación. Por ejemplo, elegimos el "track 1", le asignamos un nombre (nos aseguramos de que en la casilla "C" tenemos seleccionado un tipo de pista MIDI). A continuación, debemos seleccionar el canal y salida MIDI para la pista. En la casilla "Chn" seleccionaremos el canal MIDI 1 (Disponemos de hasta 16 canales diferentes), que será el mismo que tenga el teclado maestro MIDI. En la casilla "Output" debemos seleccionar el dispositivo de salida MIDI: si se dispone de una tarjeta compatible con Sound Blaster elegiremos "SB... (MIDI Out) ...", si es otro modelo, su salida MIDI. Si por el contrario vamos a utilizar una fuente externa de sonido, determinaremos la salida de este módulo que posiblemente llevará instalado una interfaz MPU, y pondrá seguramente "Maestro MPU-401" (Fig. 3).

Para terminar nuestra preparación, debemos seleccionar el sonido y ajustar el volumen de la pista. Esta operación la podemos realizar de dos formas diferentes. Por un lado, podemos seleccionar valores en el apartado "Track Info" de la ventana de arreglo (ver figura 3).

Colocamos un "1" en la casilla "bank" (esta casilla se refiere a los bancos de sonidos, generalmente dos: 1 y 2). En La casilla "prg" (programa), colocaremos el número del sonido que queramos. La lista de sonidos empieza en el 2,



Descripción de "Track info" de la ventana de arreglo.

así que un sonido de piano que en la tabla corresponde al primero (1), tendríamos que activarlo colocando en esta casilla un "2". Para el sonido segundo, colocaríamos un "3" y así sucesivamente. Luego, asignaremos un volumen para esa pista en la casilla "Volume". El volumen máximo es el 127 y el mínimo el 0.

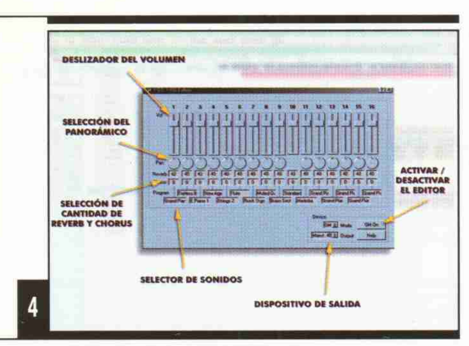
Otra forma de ajustar estos valores es mediante el Editor GM/GS/XG.

EDITOR MIDI

Para desplegar el editor GM/GS/XG pulsamos CTRL + Y o en "GM/GS/XG Editor" en el menú "Edit". Nos encontraremos ante una especie de mesa de mezclas de 16 pistas (una para cada canal MIDI). Antes de nada, en "Device", situado en la esquina inferior derecha, debemos seleccionar el dispositivo de salida como hicimos anteriormente y el modo de reproducción: GM (General MIDI), GS (Estándar de

NOTA

Si usamos una fuente de sonido externa como un módulo de sonido, debemos asegurarnos de que es compatible con General MIDI.



Descripción del editor MIDI GM/GS/XG.

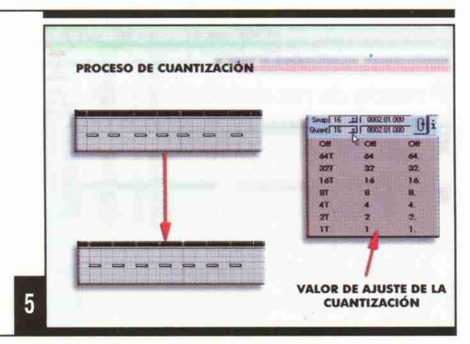
Roland) y XG (Estándar de Yamaha). Seleccionamos GM o el correspondiente al instrumento.

Para elegir el instrumento, pulsamos con el ratón en la casilla de valores de "program" de la primera pista (canal MIDI 1). En "Pan" ajustamos el panorámico y en el deslizador "Vol" el volumen de la pista. Además, podemos activar un efecto de reverberación y de *chorus* (Fig. 4).

GRABAR

Podemos seleccionar el tipo de clic del metrónomo en "Options / Metronome" entre el "beep" del ordenador o un sonido del dispositivo externo. También aquí se puede cambiar la cuenta atrás en compases antes de empezar a grabar en la casilla "precount" en número de compases "bars".

Coloquemos el localizador izquierdo de la ventana de arreglo en el comienzo de la grabación y si no sabemos hasta dónde grabaremos, desplazamos el localizador derecho hacia delante muy lejos para no quedarnos cortos. Pulsamos el botón de grabación o la tecla "*" del teclado numérico y ¡a tocar!



Proceso del ajuste de valores en la cuantización.

MÉTODOS DE GRABACIÓN

Podemos reemplazar (Replace) o mezclar (Overdub) las grabaciones en una misma pista activando o desactivando la casilla "Rec Mode" en la barra de transporte.

GRABANDO OTROS EVENTOS MIDI

Podemos también grabar cualquier otro elemento MIDI que no sea una nota, como por ejemplo, el pedal de sustain o la rueda de inflexión de tono al mismo tiempo que tocamos o de forma independiente. Para esto último, sólo tenemos que activar el modo de grabación "mezclar" (Overdub) y grabar. Oiremos la música mientras añadimos los nuevos eventos MIDI.

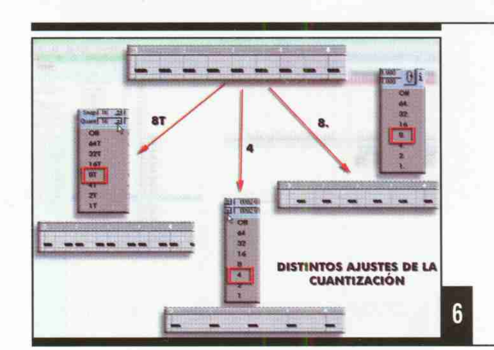
Podemos activar un ciclo de grabación y grabar en un espacio determinado de una forma continua, sumando notas a las ya existentes.

CUANTIZACIÓN

Es prácticamente imposible que la ejecución humana llegue a coincidir las notas en la medida exacta del compás. Por ello, quizás el procedimiento más importante a la hora de editar una grabación sea aplicar la cuantización.

Sin embargo, puede que no queramos ajustar nuestra grabación para que la reproducción no suene tan mecánica. Pero cuando utilicemos ritmos o un gran volumen de pistas simultáneas es imprescindible aplicar este procedimiento (Fig. 5).

Los valores de cuantización son los que utilizamos para indicarle a Cubase las posiciones exactas hacia donde mover las notas. Por ejemplo, un valor de 4 indicará una cuantización de corcheas. Hay varios métodos de cuantización que podemos activar en el menú "Functions": sobre cuantizar (Over Quantize), cuantización interactiva (Interactive Quantize), cuantización automática (Automatic Quantize), cuantización por ranuras (Groove Quantize) y cuantización analítica (Analytic Quantize). Lo más eficaz es utilizar una combinación de varias de ellas. Por



Diferentes ajustes de cuantización.

ejemplo, empezar realizando una "sobre cuantización" y si el resultado no es el óptimo aplicar posteriormente un "cuantización interactiva". En todo caso, el resultado debe ajustarse sin falta a nuestras necesidades. Hay que probar siempre con diferentes medidas de cuantización. Hay que tener en cuenta que esta función actúa siempre sobre las notas seleccionadas. En caso de que no haya ninguna, afectará a toda la pista (Fig. 6).

Una vez realizada la cuantización podemos congelarla con la opción "Functions / Freeze Quantize" para poder aplicar otra cuantización a la existente. También podemos volver a los valores iniciales antes de la cuantización en cualquier momento (incluso después de la grabación) con la opción "Functions / Undo Quantize".

Básicamente, ya estamos preparados para grabar y editar nuestro tema en este secuenciador. Sus posibilidades son enormes y sólo con la investigación y el uso se puede sacar el mayor rendimiento posible a esta herramienta.



DEFINICIÓN

▸ CUANTIZAR

Cuantizar no es más que mover las notas grabadas a posiciones exactas dentro del compás.



En el próximo número...

... conoceremos el que quizás sea el mejor editor de audio del mercado, el Sound Forge.

Desde los primeros juegos para PC a los **arcades**

Los arcades de plataformas han sido los preferidos en las recreativas de siempre, consolas y en los primeros ordenadores domésticos. El motivo principal de este éxito fue, sin duda, la riqueza visual, la incesante acción que proporcionaban y los fantásticos mundos que rodeaban a nuestro personaje.

Un juego de plataforma se caracteriza, principalmente, porque el decorado y los personajes se representan en una vista lateral. Generalmente, los escenarios donde transcurre la acción están constituidos por un fondo y decorados formados por: objetos de adorno, el suelo, el techo y las paredes. Son estos últimos elementos los que utilizan los personajes del juego, como plataformas para andar, correr o saltar.

En este número hablaremos un poco de lo que fueron y son en la actualidad estos juegos en PC, qué técnicas utilizaban y cómo evolucionaron.

LOS PRIMEROS JUEGOS EN 16 BITS PARA EL PC

Los primeros juegos que podíamos ver en un PC eran versiones de juegos existentes en sistemas de 8 bits como Spectrum 128 o MSX. Se disponía de 16 colores con tarjetas de vídeo EGA y, aunque los juegos eran exactamente iguales, carecían de un mejor atractivo visual. Prácticamente fue la propia Microsoft la que fomentó el uso del PC (clónicos del sistema IBM PC) para los desarrolladores con su simulador de vuelo *Flight Simulator*.

Pronto se empezaron a vender los compatibles PC en tiendas de electrodomésticos y aparecieron juegos exclusivos que abrieron el camino al futuro lúdico de este sistema.

Por aquel entonces, se establecieron en los PC las aventuras gráficas, sobre todo con la aparición de las tarjetas de vídeo VGA que permitían hasta 256 colores en pantalla y resoluciones más que aceptables. Cabe resaltar: las creadas por la empresa Lucas Arts como *Maniac mansión, Indiana Jones* o el clásico *Monkey Island* (1990); las producciones de Sierra, como la serie *Gabriel Knight* o el legendario *King Quest* (1984) de Roberta Williams; y la fantástica aventura *Alone in the dark* de Infogrames (1992), que introdujo la técnica de mezclar elementos 3D sobre escenarios 2D prerenderizados. Destaca también el triunfo de la Aventura Gráfica en España de la mano de Opera Soft con *La abadía del crimen*. (1989)

Gracias a los sistemas de almacenamiento que tenían los PC, estos juegos disponían de guiones extensos e infinidad de



¿SABÍAS QUE...?

La serie Comander Keen, (primeros juegos de ID Software en 1990) nació de la mano de John Carmack, quien pretendía emular la suavidad del scroll de pantalla del Mario Bros para Nintendo en el monitor del PC. Sin embargo, se encontró con el problema de la velocidad en las tarjetas de entonces y tuvo que redescubrir un sistema de impresión de tiles nuevo, creando su propio motor de scroll lateral para PC.



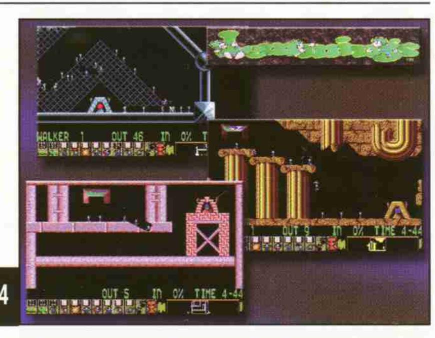
Another World y Flashback fueron los juegos de plataforma más laureados.



ID Software nació con la serie Commader Keen. Paralelamente, la veterana Opera Soft española triunfaba con Livingstone Supongo.



La serie Sims y Civilitation fueron los juegos de estrategia bandera a comienzos del PC.


4

El más auténtico y original juego de la historia, el clásico Lemmings.


5

Algunas versiones de recreativas tuvieron gran éxito para PC.


6

Con Duke Nukem (Apogee, 1991) empezaban las aventuras del "duque".


7

El primer shoot´em up 3D de la historia, HoverTank. Pero fue con Wolsfenstein 3D cuando realmente se consagró ID Software.

gráficos. Sin embargo, los que seguían triunfando eran los de plataformas, eternos y fantásticos. Para PC se realizaron verdaderas joyas como la versión de *Prince of Persia* (1990), precursor del realismo en el movimiento de los personajes.

Para los primeros 4 colores (CGA) encontramos de nuevo a Opera Soft con *Livingstone Supongo* y, ya con 16 colores (EGA) aparecieron juegos de talla como la serie *Risk Dangerous* (*Core* 1989). Y no podemos dejar de hablar de los que quizás sean de los mejores juegos para PC de la primera época: nos referimos a *Another World* y su continuación *FlashBack* de Delphine Software. El primero de ellos (1991) destaca por su increíble ambientación, creada por un sistema gráfico único en su época, en el cual se disimulaba la poca cantidad de colores en pantalla disponibles por entonces con técnicas de dibujo animado muy estilizados. La continuación, Flashback (1992), invadió los comentarios de todo el mundo: unos gráficos fascinantes llenos de detalle, unos movimientos realistas y una jugabilidad cargada de estrategia. Recordaba mucho al *Prince of Persia* y quizás sea el más versionado para otras plataformas.

Un género afloró con más fuerza en la época del PC: la estrategia. Empezamos con el innovador juego de ajedrez *Battle Chess* (1989) de InterPlay, en donde las piezas estaban dotadas de movimiento y se peleaban entre sí. Pero la estrategia pura nació de la mano de Maxis con la saga *Sims*: *Sim City* (1989), *Sim Earth* (1990), *Sim Farm* (1993). Del señor Sid Meier nace uno de los juegos de estrategia por turnos más famosos, *Civilization* (Microprose, 1991), donde controlas toda la historia de la humanidad. Sin embargo, en esta época apareció en el mercado el juego más original de la historia: *Lemmigns* (*Psygnosis*, 1991). En este juego debes llevar a un grupo de pequeños seres hacia la puerta de salida de cada nivel. Poseía unos movimientos suavísimos y perfectos, lo que permitía distinguir perfectamente todas las acciones a pesar del reducido tamaño de los personajes.

En general, entre los años 84 y 94 el género que más abundaba era el arcade y aparecieron infinidad de títulos con calidad suficiente para convertirse en clásicos. Versiones de recreativas como el *Pang, Pac Man, Tetris, Arkanoid* (1987), *Double Dragon, Paper Boy* y el fantástico *Golden Axe*. Apareció por entonces también una conversión para PC de la famosa serie *Dragon´s Lair* que introdujo un nuevo sistema de jugabilidad.

NOTA

No fue, en realidad, Wolfenstein 3D el primer shoot´em up 3D que existió, aunque sí el más conocido mundialmente.
Anteriormente a la publicación de este juego, ID Software publicó en Abril 1991 HoverTank 3D. Fue el primer juego 3D que realizó John Carmack. Después, y junto a John Romero (quién inventó la técnica del mapeado de texturas), realizaron Catacombs 3D (finales de 1991), donde aplicaron su nueva técnica, añadiendo al juego paredes de ladrillos y hierba al suelo. No se quedaron ahí, ya que quisieron que el jugador participase más en el juego y colocaron delante de la cámara un brazo con un arma. Lo demás ya es historia.

En el próximo número...

... empezaremos a comentar cada uno de los géneros de juegos para PC individualmente. Hablaremos de los juegos más significativos de los arcades de plataformas y de su técnica.

Cuestionario Videojuegos 6

Preguntas

1. **En la frase "Mi coche es verde", ¿cómo puedo eliminar la palabra "es" y quedarnos con la frase "Mi coche verde" en Blitz3D?**

2. **Describe la instrucción que nos permite detectar, por ejemplo, el cursor del ratón con un botón del menú del juego.**

3. **Señala las partes que componen la presentación de un juego, en qué orden van y cuál de ellas es estrictamente necesaria.**

4. **¿En qué partes se puede dividir el menú principal de un juego?**

5. **¿Cómo se puede crear más o menos espacio entre las letras que componen una fuente en el programa Font Creator?**

6. **¿Qué formato de objeto utilizaremos para las edificaciones y cuál para la bionave de combate?**

7. **Hemos grabado un sonido con el GoldWave en nuestro ordenador. Al reproducirlo oímos que suena demasiado bajo. ¿Cómo podemos aumentar el sonido al máximo manteniendo su calidad?**

8. **¿Cómo podemos crear con GoldWave un sonido propio sin obtenerlo del exterior?**

9. **Queremos seleccionar un sonido y aumentar el volumen de la pista en CUBASE. ¿De qué dos formas diferentes podemos realizar esta operación?**

10. **Queremos ajustar las notas de una grabación a las posiciones exactas dentro de un compás en CUBASE. ¿Qué técnica debemos utilizar?**

Respuestas al cuestionario 5

▷ **1.**

```
Type nave_enemiga
   Field x,y,z
   Field estado
   Field grafico
End Type
```

▷ **2.**

```
If Windowed3D() and GfxMode3DExists(800,600,16)
   Graphics3D 800,600,2
Endif
```

▷ **3.** Colocando el efecto de niebla con un rango de visión anterior al de la cámara.

▷ **4.** Utilizando el sistema de tileado, consistente en repetir la textura hasta cubrir todo el terreno.

▷ **5.** Seleccionando la herramienta "Object Selector" y pulsando sobre el borde de la figura.

▷ **6.** Rasterizar una capa significa convertirla a mapa de bits. Se utiliza para poder aplicar efectos de imagen a una capa de vectores.

▷ **7.** El nuevo dispositivo para reproducir o grabar se elige del panel "Device Controls Properties" en la opción "Device".

▷ **8.** Utilizando marcas con "Cue Points" del menú "Tools".

▷ **9.** Pulsando CRTL + T.

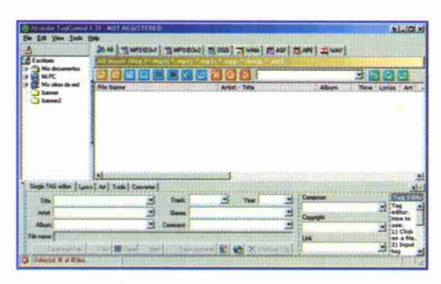
▷ **10.** Con la herramienta de línea, por ejemplo en el editor de teclas.

Contenido CD-ROM 6

▶ AUDIO

■ **Airmix Sound Automation 9.5**
Podrás mezclar música en variados formatos de audio, todo en tiempo real.

■ **Abander Tag Control 1.3**



Gracias a esta útil herramienta podrás manejar fácilmente toda tu colección de ficheros de audio.

■ **DRS 2006. The Radio Automation Software 2.1**
Potentísimo programa con el que podremos convertir nuestro PC en una auténtica máquina musical.

■ **Expresion Tone Generator 2.0**
Divertida aplicación con la cual podrás crear interesantes efectos de sonido a partir de expresiones algebraicas.

■ **Mistergenerator 1.0**
Mezcla tus archivos de audio en formato .wav fácilmente.

▶ DISEÑO 2D

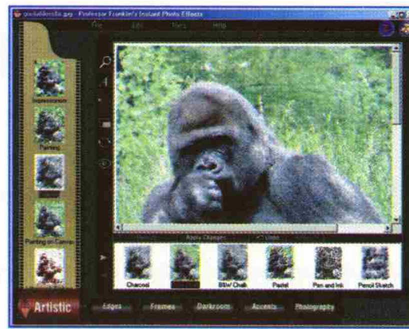
■ **Font Creator 3.0**
Con esta herramienta podrás crear y modificar fuentes True Type, para realizar la tipografía del juego.

■ **Evil Foto 3**
Completo y potente programa de catalogación de imágenes que sin duda nos será muy útil.

■ **MagicImageMarker 1.3**
Ponle marcas de agua a tus imágenes, ya sea individualmente o en grupo.

■ **Pics Print 1.16**
Imprime tus imágenes digitales fácilmente y con una gran calidad.

■ **Professor Franklins Instant Photo Effects 1.1**
Crea rápidamente sorprendentes efectos en tus imágenes.



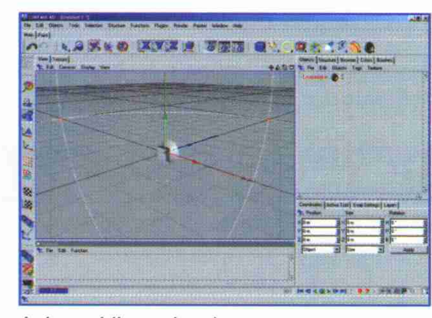
▶ DISEÑO 3D

■ **Famous 3D**
Simpático programa de animación en 3D que además podremos integrar en nuestros contactos de MSN.

■ **Pixel 3D 1.1**
Produce gráficos 3D de calidad a partir de imágenes en dos dimensiones.

■ **Cinema 4D XL7**
Gracias a este potente programa, podremos renderizar, animar personajes y crear avanzados efectos de partículas.

■ **Caligari True Space 5**



Anima tridimensionalmente con esta potente herramienta.

▶ PROGRAMACIÓN

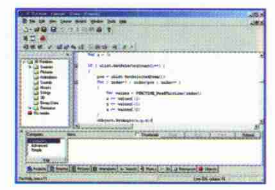
■ **3D Rad**
Entorno de desarrollo para que creemos nuestros propios juegos en tres dimensiones.

■ **DLL to Lib 1.4**
Convierte una DLL en su equivalente librería estática gracias a esta herramienta.

■ **LibCON C++ Game Programming Library 0.72 b**
API para C++ que te ayudará en la programación de tus juegos.

■ **Multimedia Fusion 1.5**
Crea juegos y otras completas aplicaciones multimedia usando este programa.

■ **Jamagic 1.2**
Completo entorno de desarrollo para crear juegos en 3D.



▶ JUEGOS

■ **Prince of Persia**
El primer juego en el que los movimientos de los personajes estaban dotados de un gran realismo.

■ **Lost Vikings**



Todo un clásico que, aun así, sigue siendo divertidísimo. Un buen juego, en definitiva.

■ **Pandemonium**
Uno de los primeros juegos en tres dimensiones que obtuvo muy buenos resultados para la época.

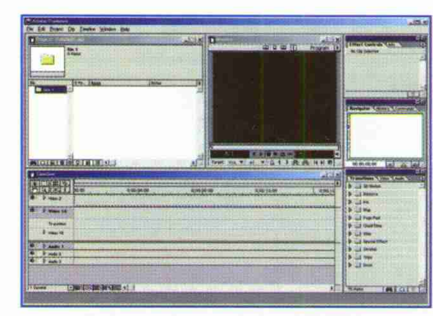
■ **Rayman 2**
Maravilloso juego, adaptación del primer título (Rayman) en 3D.

■ **Zone of Fighters**
Nueva versión de nuestro juego.

▶ VÍDEO

■ **VC Video Library 1.1**
Ten bien ordenada y clasificada tu colección de vídeos gracias a este catalogador.

■ **Adobe Premiere 6.0**



Demo del excelente programa de Adobe, uno de los más utilizados para editar vídeo.

▶ EXTRAS

En este apartado encontrarás todos los ejemplos de los que hablamos en el coleccionable, para que no pierdas detalle.